



# Apache Spark

A FAST OPEN SOURCE COMPUTE ENGINE FOR BIG DATA

by Yiqi Liu, April, 2016

# THE PROBLEM — BIG DATA

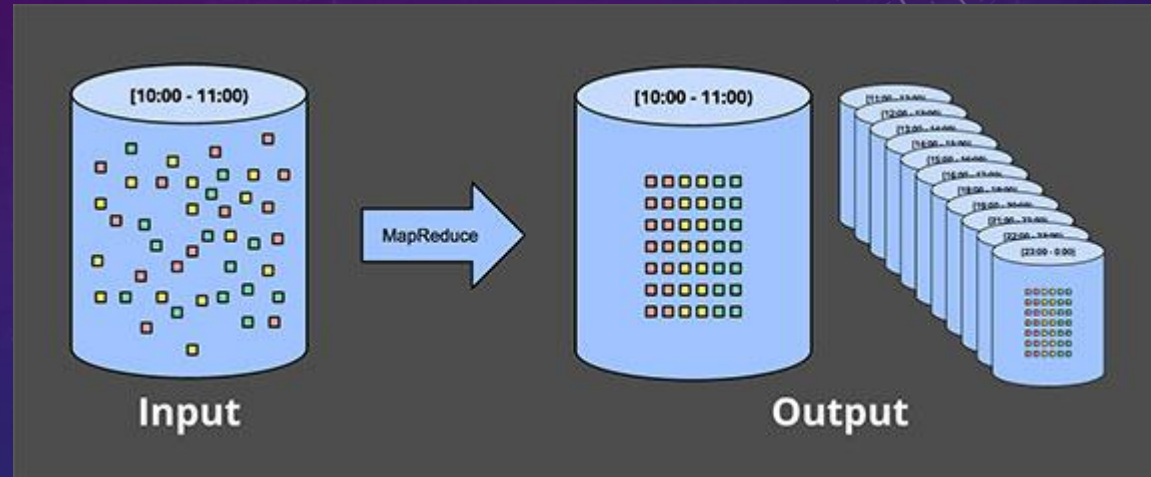
- Scenarios:
  - aerodynamic simulation
  - Prediction (e.g. Netflix)
  - Unmanned vehicle



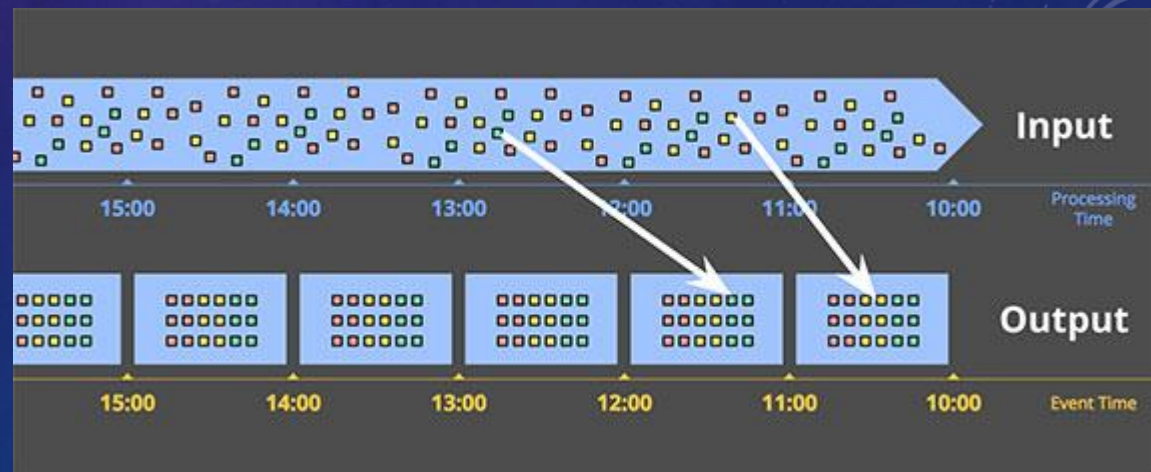
# APPROACHES

## Batch - Microbatch - Realtime

Data collected upfront

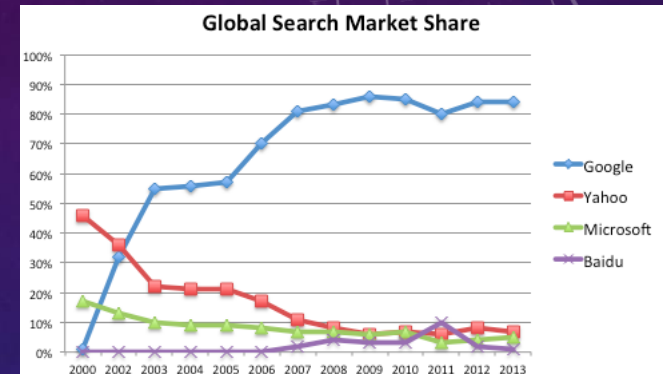


Data processed as collected



# THE CONTEXT

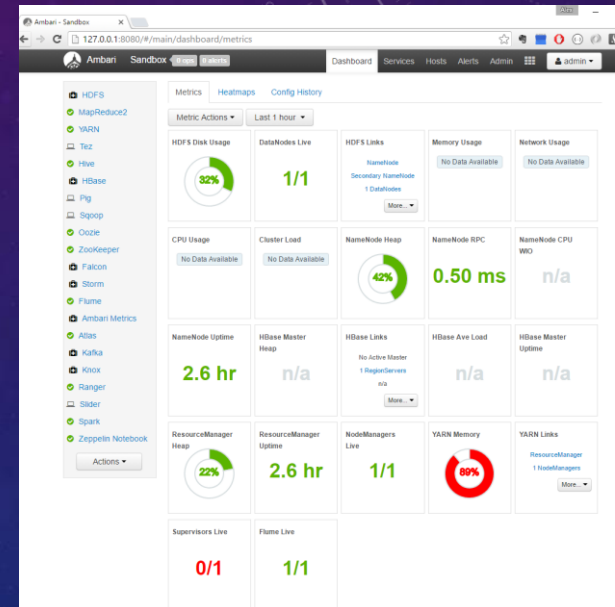
- Google vs Yahoo in early days
  - Web crawler
  - 2003: Google File System paper released
- MapReduce + Hadoop(2004 ~ 2006)
  - 2006: Yahoo deploys 300 machine Hadoop cluster
- New technologies built on Hadoop
- Hadoop eco-system



By Mike Tekula Published [January 21, 2014](#) in the [Marketing](#), [SEO](#) categories



# HADOOP ECO-SYSTEM



Screen shot from Hortonworks Data Platform

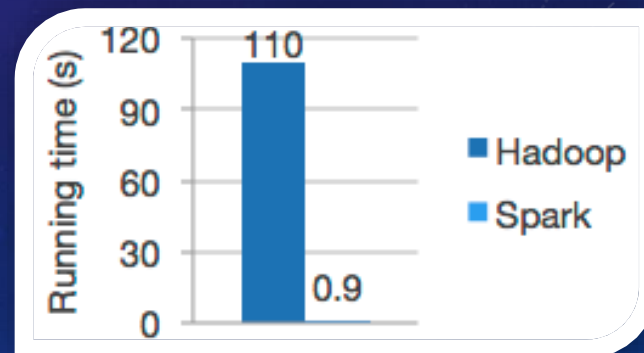
<http://techiekhannotes.blogspot.ca/2015/08/hadoop-ecosystem.html>



[https://en.wikipedia.org/wiki/Apache\\_Hadoop](https://en.wikipedia.org/wiki/Apache_Hadoop)

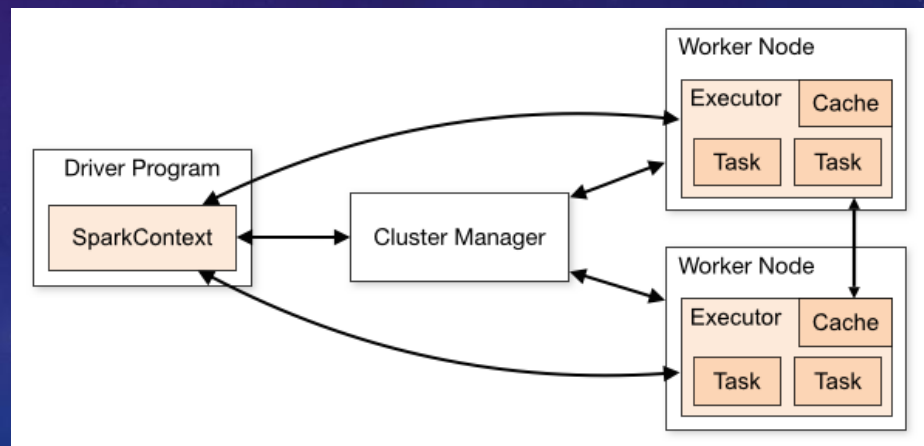
# SPARK

- Fast — Quick iterations (ML, AD-HOC)
- Easy to use (Data Scientists)
- 1 > 3: Flexible (Batch — Streaming)
- Spark vs MapReduce
- Spark vs Storm



# TYPICAL SETUP

- Network: 10 Gb/s NIC. x 3 etc.
- Starts from hundreds of nodes
- Node
  - 64 GB RAM each
  - Powerful CPU(s)
  - TBs of data



<http://spark.apache.org/docs/latest/cluster-overview.html>

# EXAMPLE – FLIGHT DELAY ANALYSIS

Based on HDP Lab

Extract (9.3 MB, 100K lines)

```
wget https://raw.githubusercontent.com/roberthryniewicz/datasets/master/airline-dataset/flights/flights.csv -O /tmp/flights.csv
```

```
%sh
```

```
# put data into HDFS
```

```
hadoop fs -put /tmp/flights.csv /tmp/airflightsdelays/
```

```
Year,Month,DayOfMonth,DayOfWeek,DepTime,CRSDepTime,ArrTime,CRSArrTime,UniqueCarrier,FlightNum,TailNum,ActualElapsedTime,CRSElapsedTime,AirTime,ArrDelay,DepDelay,Origin,Dest,Distance,TaxiIn,TaxiOut,Cancelled,CancellationCode,Diverted,CarrierDelay,WeatherDelay,NASDelay,SecurityDelay,LateAircraftDelay
2008,1,3,4,2003,1955,2211,2225,WN,335,N712SW,128,150,116,-14,8,IAD,TPA,810,4,8,0,,0,NA,NA,NA,NA,NA
2008,1,3,4,754,735,1002,1000,WN,3231,N772SW,128,145,113,2,19,IAD,TPA,810,5,10,0,,0,NA,NA,NA,NA,NA
```

```
// Create a DataFrame from datasetsval
```

```
df = sqlContext.read
```

```
  .format("com.databricks.spark.csv")
```

```
  .option("header","true")
```

```
  .option("inferSchema","true") // Type inference
```

```
  .load("/tmp/airflightsdelays/")
```



# EXAMPLE (Cont.)

## Transform and Analysis

```
import org.apache.spark.sql.functions.udf

val isDelayedUDF = udf((time: String) =>
  if (time == "NA") 0 else if (time.toInt > 15) 1 else 0)
val updatedDF = df.select($"Year", $"Month", ... ,
  isDelayedUDF($"DepDelay").alias("IsDelayed")).cache

updatedDF.agg((sum("IsDelayed") * 100 / count("DepDelay"))
  .alias("Percentage of Delayed Flights")).show
```

Year	Month	DayOfMonth	DayOfWeek	CRSDepTime	UniqueCarrier	FlightNum	DepDelay	Origin	Dest	TaxiIn	TaxiOut	Distance	IsDelayed
2008	1	3	4	1955	WN	335	8	IAD	TPA	4	8	810	0
2008	1	3	4	735	WN	3231	19	IAD	TPA	5	10	810	1
2008	1	3	4	620	WN	448	8	IND	BWI	3	17	515	0

```
updatedDF.agg((sum("IsDelayed") * 100 / count("DepDelay"))
  .alias("Percentage of Delayed Flights")).show
```

Percentage of Delayed Flights
19.587

Compute time: 26s

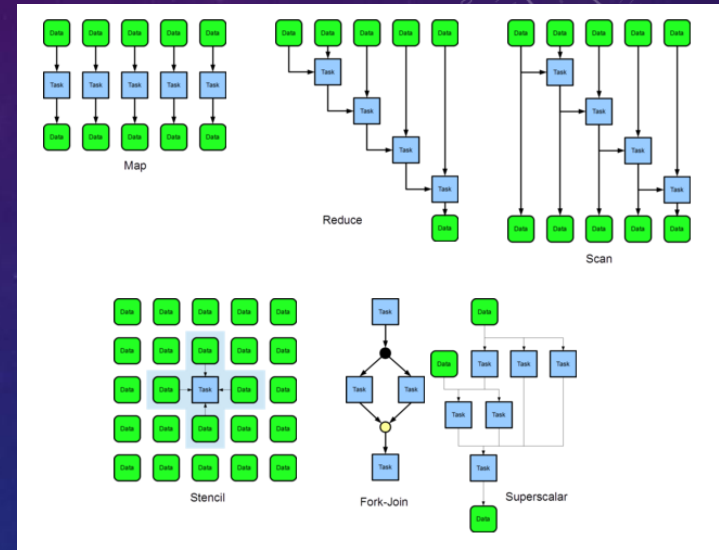
Human time: ?

# SCALA

- From Novice to Master
- Efficient
  - Fast prototyping
  - Static typed
  - Compiled
- Scalability (functional, parallel)
  - `list.par.map(_ + 42)`

# COMMON CONCEPTS

- Parallelism
- Divide and conquer
- Partitioning (Grain size)
- Communication overhead
- Data/Task Dependencies
- Manager - Worker
- Fork – Join, reduce, ...
- MPI
- ...
- 1 line of code = \$1 million ?



Parallel Control Patterns

Chris Szalwinski, Parallel Programming. Retrieved 04/10/16  
<https://scs.senecac.on.ca/~gpu621/pages/content/patte.html>